Segmentacja obrazów cyfrowych z zastosowaniem teorii grafów - wstęp

autor: Łukasz Chlebda

Temat pracy:

Aplikacja do segmentacji obrazów cyfrowych z zastosowaniem teorii grafów

Promotor:

dr hab. inż. Andrzej Leśniak, prof. AGH

Plan prezentacji:

- Techniki segmentacji obrazów cyfrowych
- Zarys teorii grafów
- Wykorzystane algorytmy grafowe
- Obraz cyfrowy jako graf
- Opis metod grafowych
- Napotkane problemy
- Wstępne wyniki

Segmentacja obrazów cyfrowych

Segmentacja polega na podziale obrazu na fragmenty odpowiadające poszczególnym widocznym na obrazie obiektom.

Umożliwia wydzielenie obszarów obrazu spełniających pewne kryteria jednorodności (np.. kolor obszaru, faktura, poziom jasności).

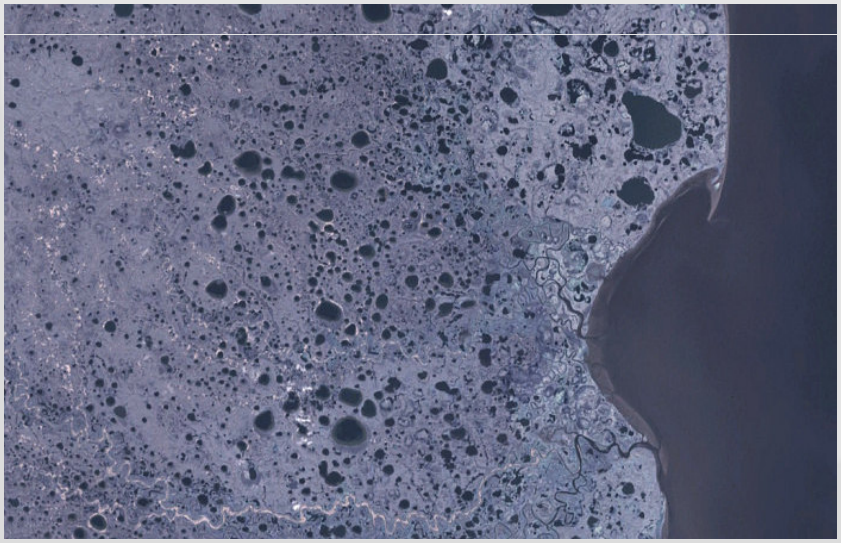
Najbardziej powszechne techniki segmentacji:

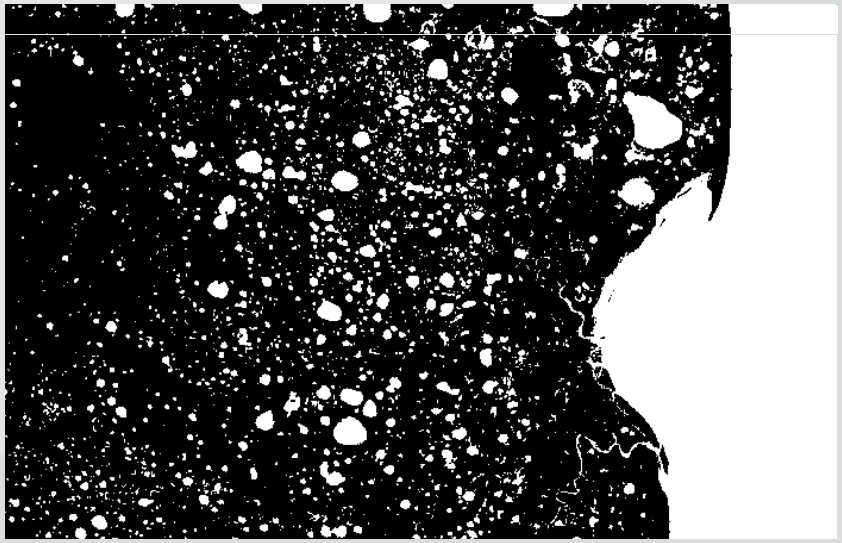
- segmentacja przez podział obszaru
- segmentacja metodą wykrywanie krawędzi
- segmentacja przez rozrost obszaru

Segmentacja przez podział obszaru

Wartość jasności każdego elementu obrazu jest porównywana z wartością progową, po czym element jest przydzielany do jednej z dwóch kategorii. Wyboru wartości progowej dokonuje się zazwyczaj na podstawie histogramu.

Przykład – wieczna zmarzlina – dla progu równego 100:

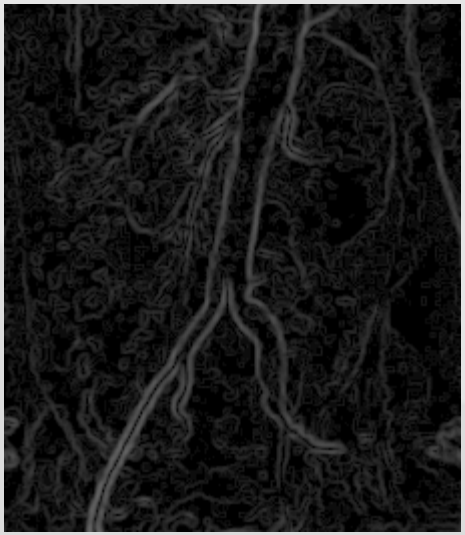




Segmentacja metodą wykrywania krawędzi

Przy tej metodzie są wyszukiwane krawędzie między obszarami – najczęściej za pomocą operatorów gradientowych. Następnie najczęściej jest wykonywana operacja progowania.

Przykład – obrazek został przetworzony najpierw operatorami Sobela, później nałożono na niego próg progowania 50:

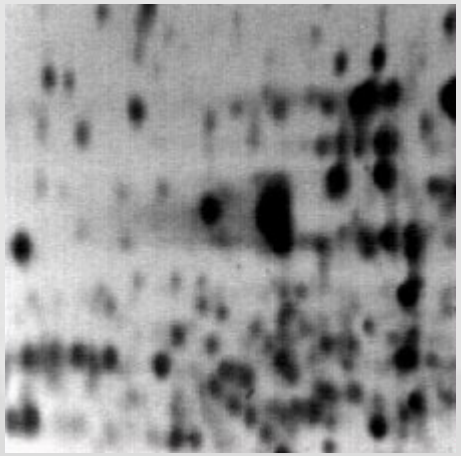


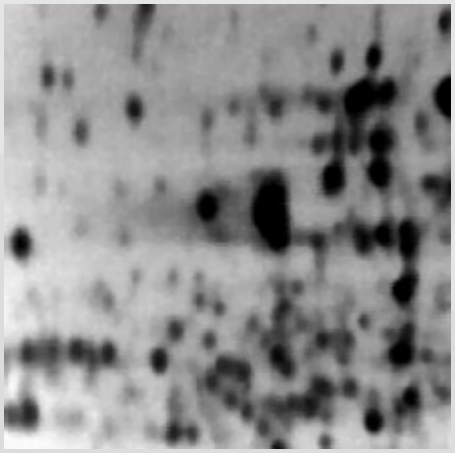
Segmentacja przez rozrost obszaru

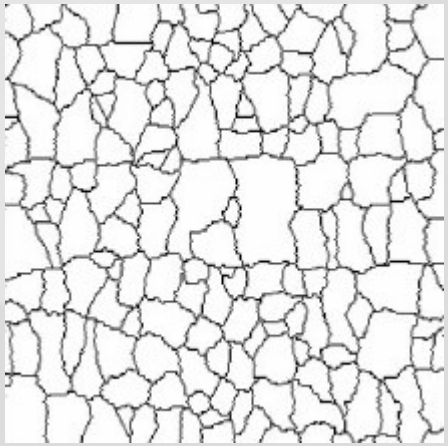
W metodzie tej poszukuje się grup elementów o zbliżonej jasności.

Przykładowo – segmentacja morfologiczna – algorytm linii działu wodnego (LDW).

Przykład – elektrofereza żelu – najpierw obraz przefiltrowano (otwarcie + zamknięcie) celem uniknięcia przesegmentowania, by następnie zastosować algorytm LDW dla obrazka przefiltrowanego:





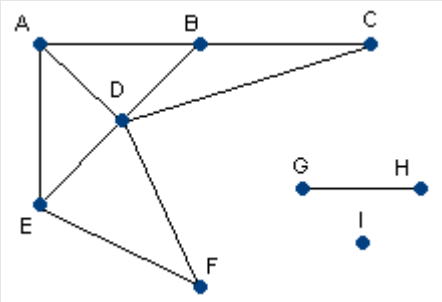


Zarys teorii grafów

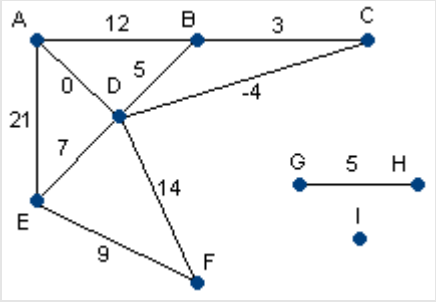
- Parę G = (V(G), E(G)) (lub w skrócie G=(V, E)) nazywamy **grafem**, jeżeli V(G) jest zbiorem skończonym $E(G) \subset V(G) \times V(G)$
- Elementy V(G) nazywamy **wierzchołkami** grafu G, a V(G) **zbiorem wierzchołków**.
- Elementy E(G) nazywamy **krawędziami** grafu G, a E(G) **zbiorem krawędzi**.
- Graf jest **spójny** jeżeli dla dowolnej pary wierzchołków istnieje droga łącząca te wierzchołki.
- **Cykl** – to zamknięta droga, w której wszystkie krawędzie są różne

Rodzaje grafów

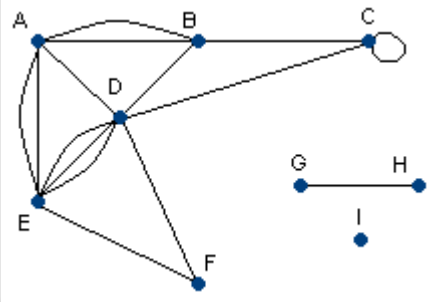
Graf prosty



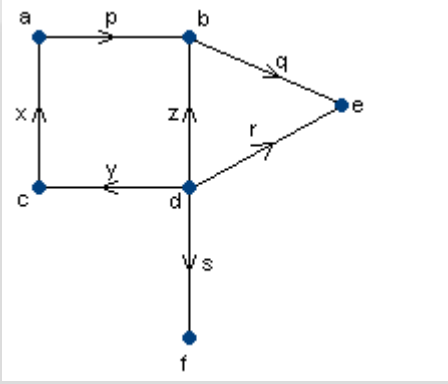
Graf z wagami



Multigraf (z krawędziami wielokrotnymi i pętlami)



Digraf (graf skierowany)



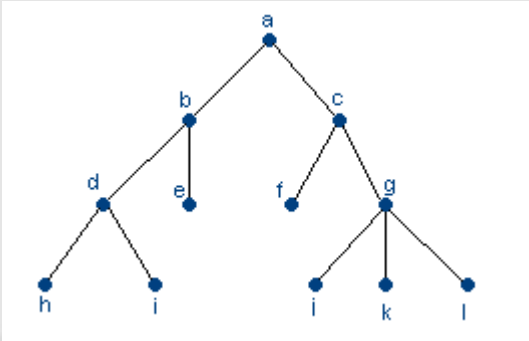
Drzewa

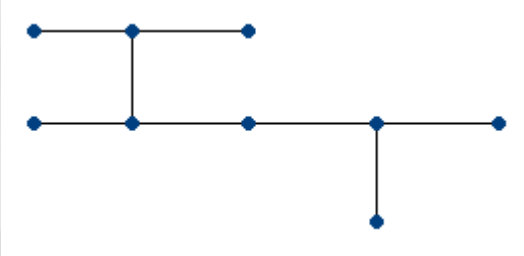
Drzewem nazywamy graf spójny i acykliczny.

Drzewem spinającym nazywamy minimalny (o minimalnej liczbie krawędzi) podgraf łączący wszystkie wierzchołki grafu G.

Minimalne drzewo spinające grafu G to drzewo spinające o minimalnej wadze.

Przykłady drzew:





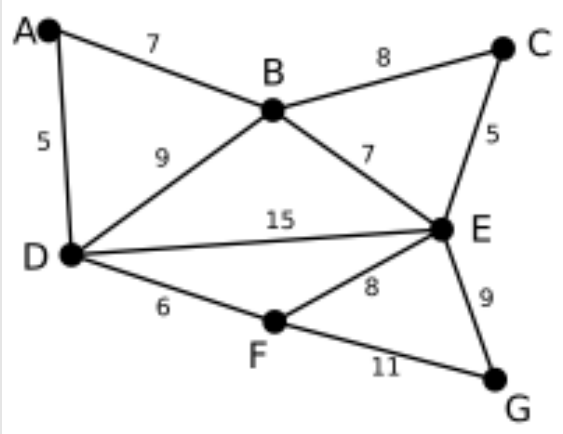
Algorytmy grafowe

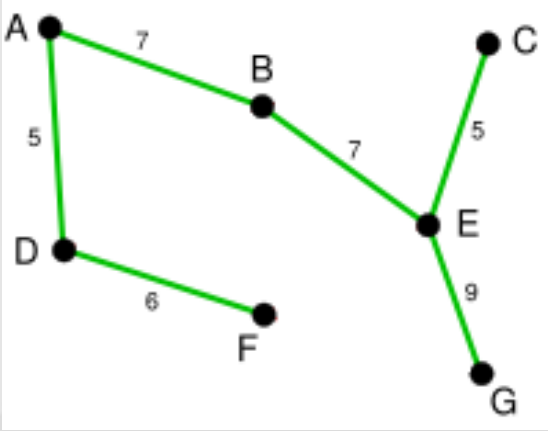
Algorytm Dijkstry - wyznacza minimalną drogę między wierzchołkiem u i każdym z pozostałych wierzchołków w skończonym digrafie z nieujemnymi wagami bez pętli i bez krawędzi wielokrotnych.

$L(i)/pred(i)$	A	B	C	D	E	F	G	H	I
A	$0/\varnothing$	∞/A	$2/A$	$\boxed{1/A}$	∞/A	∞/A	∞/A	∞/A	∞/A
AD		$4/D$	$\boxed{2/A}$		$4/D$	∞/A	∞/A	∞/A	∞/A
ADC		$\boxed{4/D}$			$4/D$	∞/A	∞/A	$4/C$	∞/A
ADCB					$\boxed{4/D}$	∞/A	$6/B$	$4/C$	∞/A
ADCBE						∞/A	$6/B$	$\boxed{4/C}$	∞/A
ADCBEH						$8/H$	$\boxed{6/B}$		∞/A
ADCBEHG						$\boxed{7/G}$			∞/A
ADCBEHGF									$\boxed{\infty/A}$
{A,D,C,B,E,H,G,F,I} X = V(G)									

Algorytm Kruskala – daje w wyniku minimalne drzewo spinające grafu spójnego G.

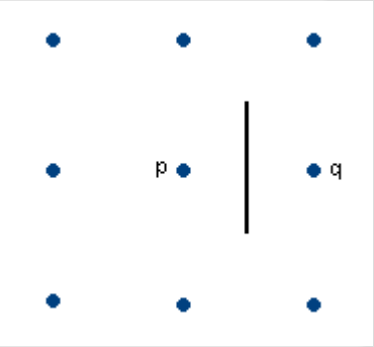
Przykład grafu oraz minimalnego drzewa spinającego tego grafu:





Obraz cyfrowy jako graf

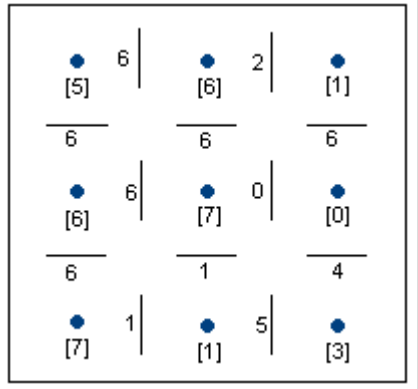
- Zakładamy, że piksele są 4 – sąsiadowi
- Krawędź grafu definiujemy jako granicę pomiędzy pikselami p i q, jeżeli są one 4 – sąsiadami.



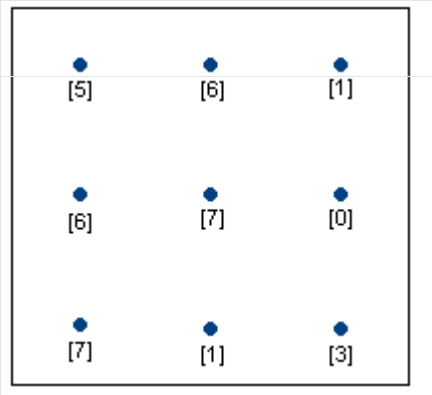
Każdej krawędzi przypisujemy wagę, którą określamy następująco:

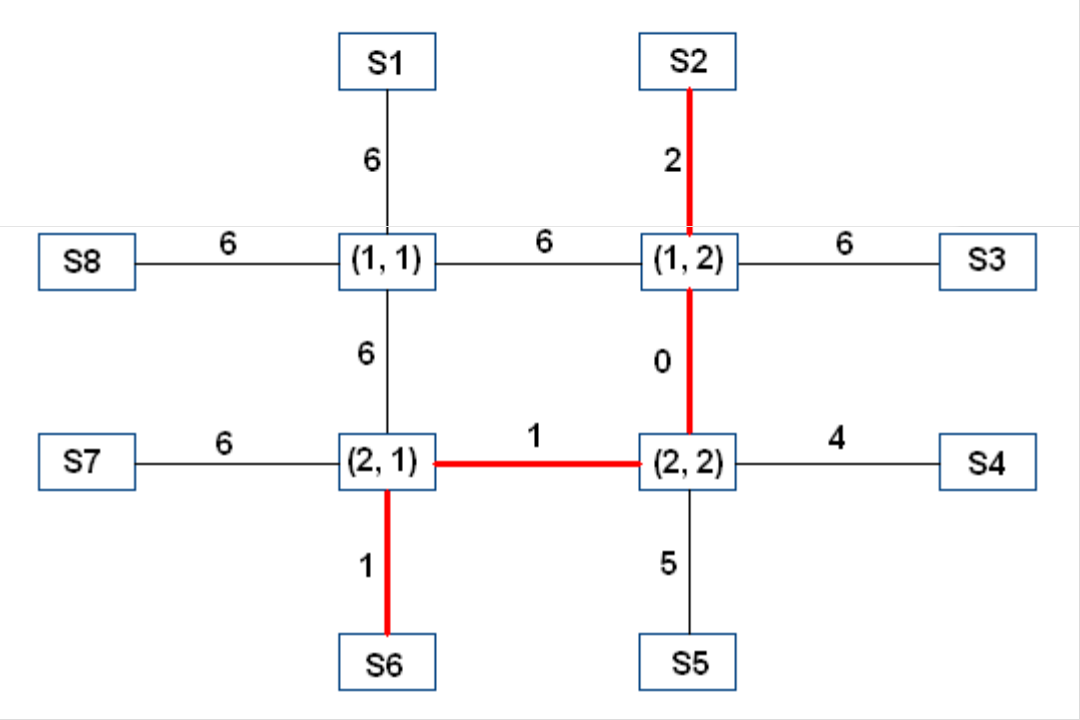
$$w(p, q) = H - | [f(p) - f(q)] |$$

gdzie H jest najwyższą wartością piksela na obrazku, a f(p) i f(q) są wartościami pikseli odpowiadających wierzchołkom p i q.



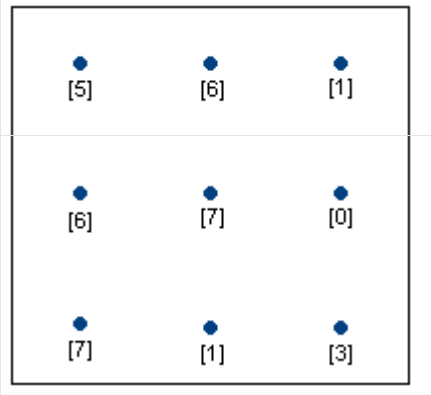
Przykład obrazu cyfrowego oraz jego reprezentacja grafowa. Na czerwono zaznaczona jest najkrótsza droga łącząca przeciwległe brzegi grafu – wyznaczona algorytmem Dijkstry.

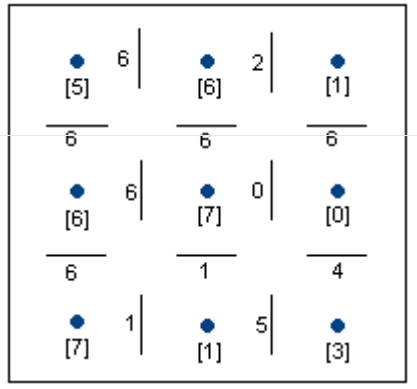


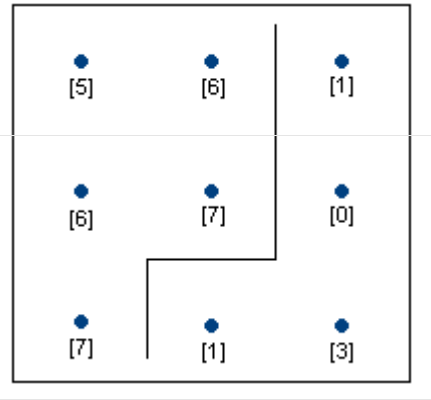


Metoda wykrywająca krawędzie

Okazuje się, że **najkrótsze drogi w grafie** odpowiadają znaczącym zmianom jasności między dwoma pikselami w obrazie – czyli po prostu krawędziom.







Metoda używana w tej pracy oparta jest na tzw. **algorytmie A***, który służy do znajdywania najkrótszej drogi pomiędzy dwoma danymi wierzchołkami grafu, jednak nie gwarantuje minimalnej drogi pomiędzy dwoma punktami.

Algorytm A*

- Opiera się na **heurystyce** - dzięki czemu jest znacznie szybszy
- **Algorytm Dijkstry** jest szczególnym przypadkiem algorytmu A* - wtedy gdy nie korzystamy z żadnej heurystyki

Heurystyka jest to metoda znajdywania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego. Używa się jej, gdy algorytm z przyczyn technicznych jest zbyt kosztowny. W przypadku algorytmu A* heurystyka jest wykorzystywana do nakierowywania pełnego algorytmu ku optymalnemu rozwiązaniu bez poświęcania jakości rozwiązania.

Parametry

- **czułość** - minimalna różnica pomiędzy dwoma pikselami 4 – sąsiadowymi, która będzie traktowana jako krawędź. Krawędzie, których wagi są mniejsze będą pomijane.
- **maski** - obrazek jest dzielony na maski o podanych przez użytkownika rozmiarach, a algorytm jest wykonywany na każdej masce oddzielnie.

Metoda segmentacji grafowej

Dość skomplikowana metoda dzieląca obraz na obszary o podobnych właściwościach – wykorzystująca m. in. **minimalne drzewa spinające**.

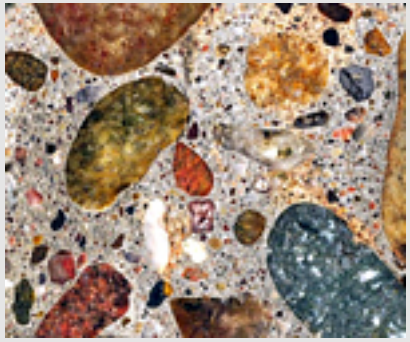
- graf jest konstruowany na takich samych zasadach, jak wcześniej
- zakładamy, że piksele są 4 lub 8 – sąsiadowe
- w metodzie tej wykorzystałem **algorytm Kruskala**
- metoda ta wymaga posortowania na początku wszystkich krawędzi ze względu na wagi – wykorzystałem do tego algorytm sortowania **quicksort**

Na podstawie pracy „*Effcient Graph-Based Image Segmentation*" autora Pedro F. Felzenszwalb z Artificial Intelligence Lab – MIT.

Napotkane problemy

- szybkość
- narzut pamięciowy

obrazek 150 x 125



18 750 wierzchołków

37 225 krawędzi

obrazek 819 x 614

1 004 299 krawędzi

502 866 wierzchołków

Jak przyspieszyłem działanie algorytmów

- w metodzie wykrywania krawędzi zamiast całego obrazka naraz – tworzone są maski
- zanim utworzony zostanie graf – sprawdzane jest, czy na danej masce w ogóle może wystąpić krawędź
- heurystyka
- quicksort
- algorytm Kruskala zamiast algorytmu Prima
- (w mojej głowie przyspieszam je także równolegle)

Przydałoby się szybciej…

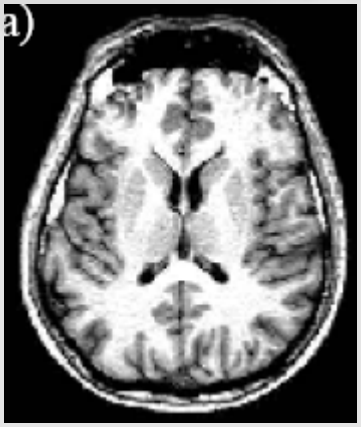
• algorytm Bernarda Chazelle – najszybszy obecnie algorytm znajdujący minimalne drzewo spinające - wykorzystujący tzw. „miękki stos”

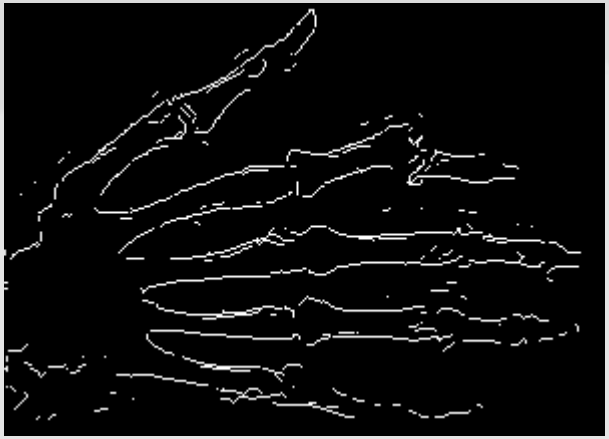
Algorytm ten jest praktycznie liniowo zależny od liczby wierzchołków oraz tzw. odwrotnej liczby Ackermana.

Nierozwiązany problem – narzut pamięciowy

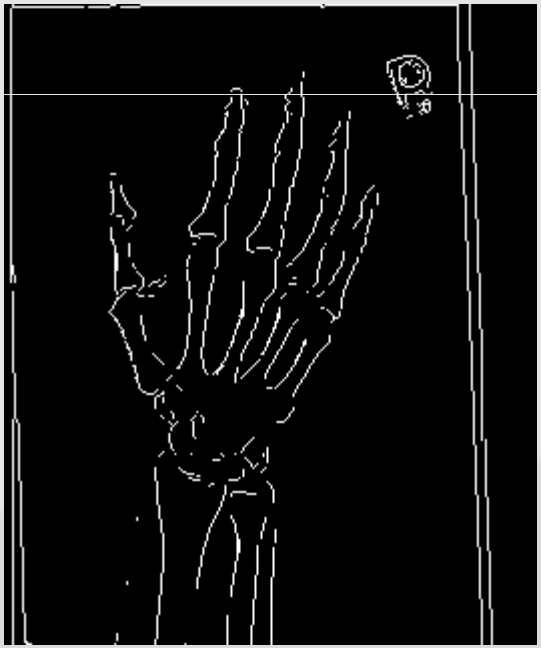
- występuje przy metodzie segmentacji grafowej
- jeżeli starczy czasu – spróbuje zaimplementować inną strukturę danych – na razie wykorzystuje dynamiczne listy wierzchołków i krawędzi powiązanych ze sobą

Przykładowe wyniki – wykrywanie krawędzi:

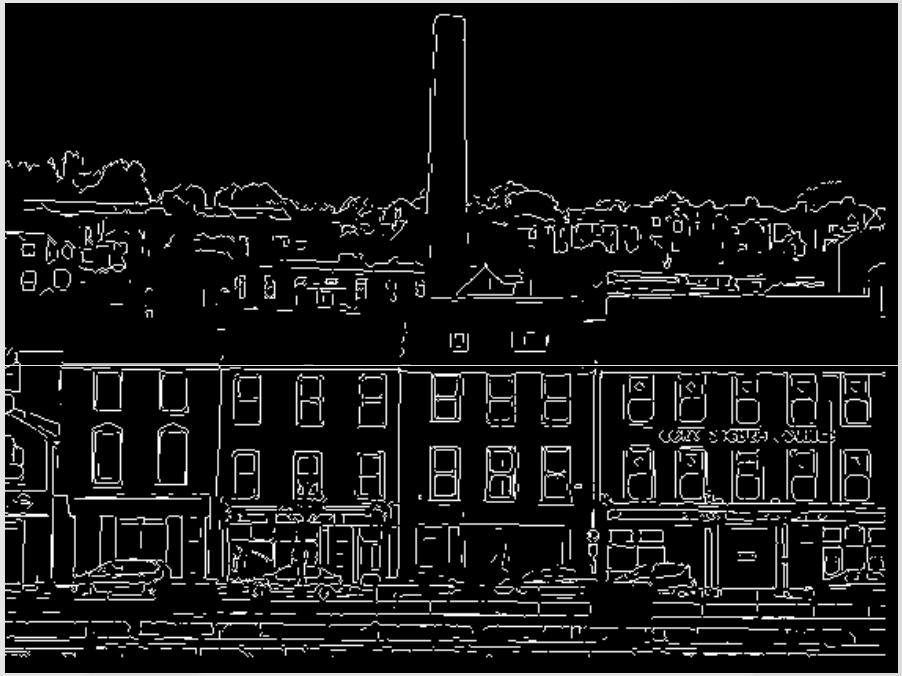




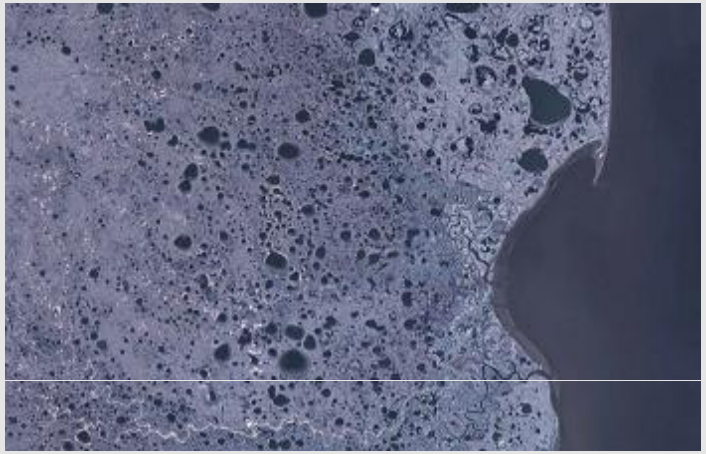

CAD

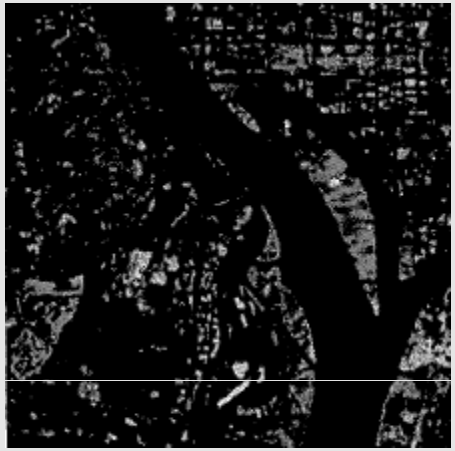


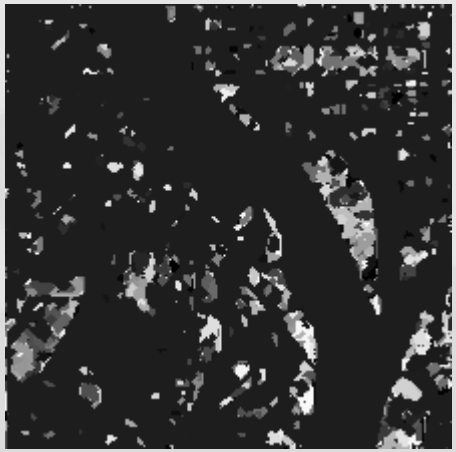

CORK ENGLISH COLLEGE
ORIENTAL RUGS CO.
www.corkenglishcollege.ie



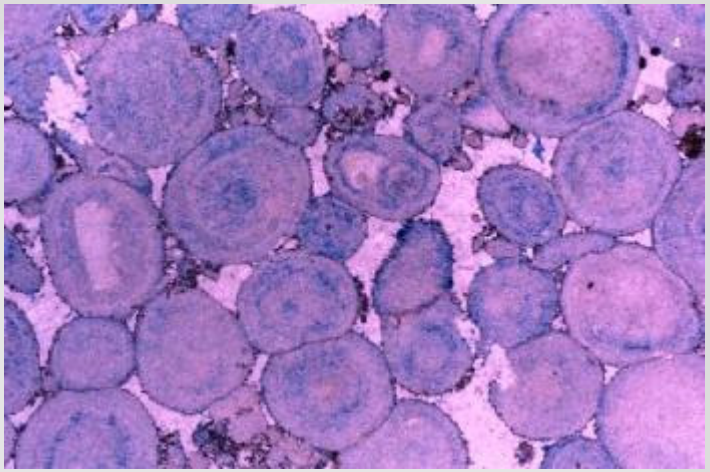
Przykładowe wyniki – segmentacja grafowa

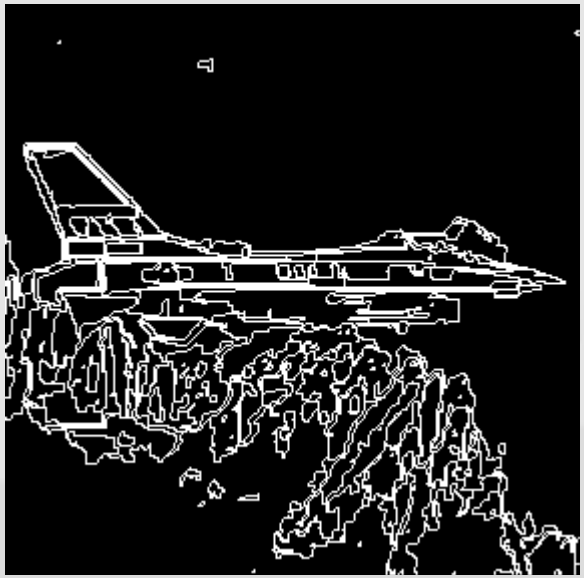


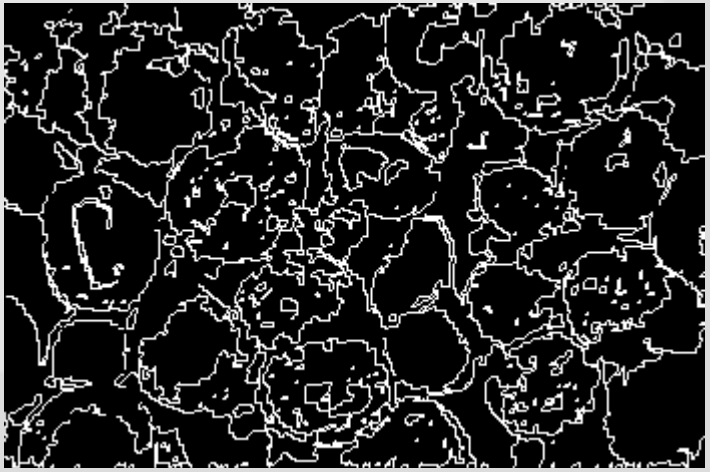


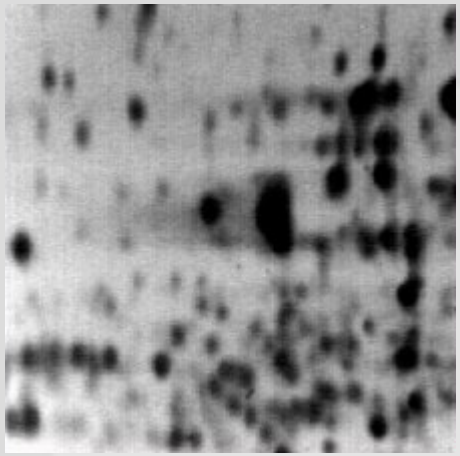


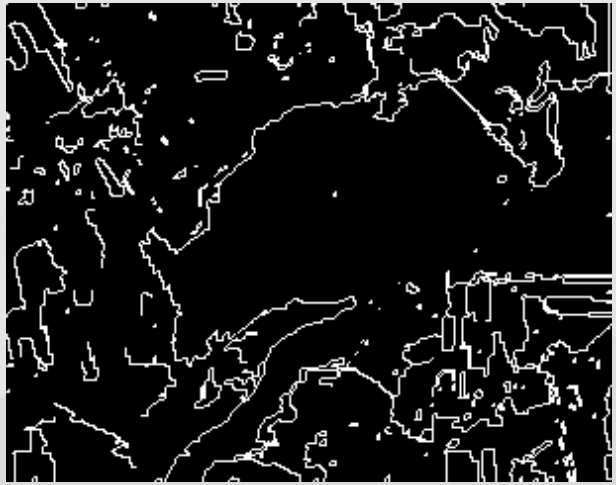

F-16
01568
U.S. AIR FORCE











Dziękuję za uwagę.